

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-162968

(P2002-162968A)

(43)公開日 平成14年6月7日(2002.6.7)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テマコード*(参考)
G 1 0 H 1/00	1 0 2	G 1 0 H 1/00	1 0 2 Z 5 D 3 7 8
1/40		1/40	

審査請求 未請求 請求項の数19 O L (全 16 頁)

(21)出願番号 特願2000-361765(P2000-361765)

(22)出願日 平成12年11月28日(2000.11.28)

(71)出願人 500521603

株式会社エム・ゾーン

東京都新宿区市谷薬王寺町58 春日マンション4階

(71)出願人 500546433

アイ・ティー・テレコム株式会社

東京都文京区音羽2丁目10番2号 音羽N Sビル

(72)発明者 大曲 史朗

広島県広島市西区古江上2-430-1 コンツェルトパーク古江403

(74)代理人 100083806

弁理士 三好 秀和 (外8名)

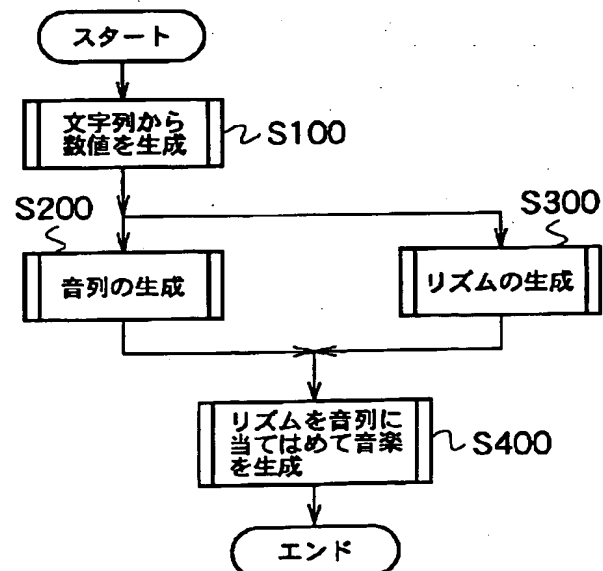
最終頁に続く

(54)【発明の名称】 音楽自動生成方法、音楽自動生成プログラム、および音楽自動生成プログラムを記録した記録媒体

(57)【要約】

【課題】 音楽を自動的に生成する技術を提供する。

【解決手段】 文字列から数値を生成するステップ(S100)と、数値から音列を生成するステップ(S200)と、数値からリズムを生成するステップ(S300)と、リズムを音列に当てはめて音楽を生成するステップ(S400)とを含む。



## 【特許請求の範囲】

【請求項 1】 平面内における所定数の座標点の X 座標値と Y 座標値を取得する X Y 座標値取得ステップと、前記所定数の座標点のそれぞれに対応する数値を取得する座標点対応数値取得ステップと、前記座標点対応数値を所定の音階数で割った余りに従って音を割り当てることによって音列を生成するステップと、を少なくとも含むことを特徴とする音楽自動生成方法。

【請求項 2】 前記音楽自動生成方法であって、さらに、前記平面内の各座標点のそれぞれに予め数値を対応づけておくステップを含むことを特徴とする請求項 1 に記載の音楽自動生成方法。

【請求項 3】 前記音楽自動生成方法であって、前記座標点対応数値取得ステップは、画像を読み込み、画像を構成する各点の色の明るさを決定する数値を前記座標点のそれぞれに対応する数値として取得することを特徴とする請求項 1 乃至 2 に記載の音楽自動生成方法。

【請求項 4】 前記音楽自動生成方法であって、前記座標点対応数値取得ステップは、前記座標点の座標値とフラクタル関数とを用いて、各座標点に対応する数値を取得することを特徴とする請求項 1 に記載の音楽自動生成方法。

【請求項 5】 前記音楽自動生成方法であって、前記平面が複素平面であり、前記座標点対応数値取得ステップは、 $Z(0) = c$   
 $Z(n) = Z(n-1)^2 + c$   
 但し、 $c$  の実数部分は前記 X 座標値、 $c$  の虚数部分は前記 Y 座標値  
 $n$  は自然数  
 で表される  $Z(n)$  の絶対値が発散するまでの演算繰り返し回数  $T$  を求めて、該演算繰り返し回数  $T$  を前記座標点対応数値とすることを特徴とする請求項 1 に記載の音楽自動生成方法。

【請求項 6】 前記音楽自動生成方法であって、前記 X Y 座標値取得ステップは、前記平面内における始点の X 座標値  $SX$  と、始点の Y 座標値  $SY$  と、X 方向の増分値  $DX$  と、Y 方向の増分値  $DY$  とを取得し、始点を第 1 の座標点とし、  
 (始点の X 座標値  $SX + m \times$  増分値  $DX$ 、始点の Y 座標値  $SY + m \times$  増分値  $DY$ ) を第  $n$  番目の座標点とする  
 ( $m, n$  は自然数) ことを特徴とする請求項 1 乃至 5 のいずれかに記載の音楽自動生成方法。

【請求項 7】 前記音楽自動生成方法であって、さらに、文字列を取得するステップと、

前記文字列を構成する各文字を文字コードに変換することによって前記文字列を文字コード列に変換するステップと、

前記文字コード列を構成する各英数字をビット列に変換することによって前記文字コード列から第 1 のビット列を生成するステップと、

前記第 1 のビット列から複数のビットを抜き出して得られる複数の第 2 のビット列のそれぞれを数値に変換することによって複数の数値  $V1, V2, V3, V4$  を生成するステップと、

前記複数の数値  $V1, V2, V3, V4$  から前記  $SX, SY, DX, DY$  を生成するステップと、を含むことを特徴とする請求項 6 に記載の音楽自動生成方法。

【請求項 8】 複素平面内における始点の X 座標値  $SX$  と、始点の Y 座標値  $SY$  と、X 方向の増分値  $DX$  と、Y 方向の増分値  $DY$  とを取得するステップと、

$$Z(0) = c$$

$$Z(n) = Z(n-1)^2 + c$$

但し、 $c$  の実数部分は始点の X 座標値  $SX$ 、 $c$  の虚数部分は始点の Y 座標値  $SY$

$n$  は自然数  
 で表される  $Z(n)$  の絶対値が発散するまでの演算繰り返し回数  $T$  を求めるステップと、

$$Z(0) = c$$

$$Z(n) = Z(n-1)^2 + c$$

但し、 $c$  の実数部分は (始点の X 座標値  $SX + m \times$  増分値  $DX$ )、 $c$  の虚数部分は (始点の Y 座標値  $SY + m \times$  増分値  $DY$ )

$m, n$  は自然数

で表される  $Z(n)$  の絶対値が発散するまでの演算繰り返し回数  $T$  を求めるステップと、

0.01 以上 0.99 以下の数値を取得して、この数値を  $X(0)$  とするステップと、

$$X(n) = \lambda * X(n-1) * (1 - X(n-1))$$

但し、 $\lambda$  は 4 未満の正の数

$n$  は自然数

で表される  $X(n)$  を順次計算して、各  $X(n)$  を算出するステップと、

前記各演算繰り返し回数  $T$  が奇数か否かと、前記  $X(n)$  と予め設定したしきい値との大小関係と、を判断するステップと、

前記奇数か否かの判断結果と、前記大小関係の判断結果とに基づいて、「ノートオン (音を鳴らす)」、「前の音を継続」または「休符」のいずれかを割り当てることによってリズムを生成するステップと、を少なくとも含むことを特徴とする音楽自動生成方法。

【請求項 9】 前記音楽自動生成方法であって、さらに、

前記数値  $V1, V2, V3, V4$  のいずれかをこれら数値が取りうる最大値で割った値に 0.01 を加えて数値

X(0)を生成するステップと、  
 $X(n) = \lambda * X(n-1) (1 - X(n-1))$   
 但し、 $\lambda$ は4未満の正の数  
 $n$ は自然数  
 で表されるX(n)を順次計算して、各X(n)を算出するステップと、  
 前記各演算繰り返し回数Tが奇数か否かと、前記X(n)と予め設定したしきい値との大小関係と、を判断するステップと、  
 前記奇数か否かの判断結果と、前記大小関係の判断結果とに基づいて、「ノートオン(音を鳴らす)」、「前の音を継続」または「休符」のいずれかを割り当てることによってリズムを生成するステップと、を少なくとも含むことを特徴とする請求項1乃至7に記載の音楽自動生成方法。  
 【請求項10】 平面内における所定数の座標点のX座標値とY座標値を取得するXY座標値取得ステップと、前記所定数の座標点のそれぞれに対応する数値を取得する座標点对応数値取得ステップと、前記座標点对応数値を所定の音階数で割った余りに従って音を割り当てることによって音列を生成するステップと、をコンピュータに実行させることを特徴とする音楽自動生成プログラム。  
 【請求項11】 前記音楽自動生成プログラムであって、さらに、前記平面内の各座標点のそれぞれに予め数値を対応づけておくステップを含むことを特徴とする請求項10に記載の音楽自動生成プログラム。  
 【請求項12】 前記音楽自動生成プログラムであって、前記座標点对応数値取得ステップは、画像を読み込み、画像を構成する各点の色の明るさを決定する数値を前記座標点のそれぞれに対応する数値として取得することを特徴とする請求項11に記載の音楽自動生成プログラム。  
 【請求項13】 前記音楽自動生成プログラムであって、前記座標点对応数値取得ステップは、前記座標点の座標値とフラクタル関数とを用いて、各座標点に対応する数値を取得することを特徴とする請求項10に記載の音楽自動生成プログラム。  
 【請求項14】 前記音楽自動生成プログラムであって、前記平面が複素平面であり、前記座標点对応数値取得ステップは、  
 $Z(0) = c$   
 $Z(n) = Z(n-1)^2 + c$   
 但し、 $c$ の実数部分は前記X座標値、 $c$ の虚数部分は前記Y座標値

$n$ は自然数  
 で表されるZ(n)の絶対値が発散するまでの演算繰り返し回数Tを求めて、該演算繰り返し回数Tを前記座標点对応数値とすることを特徴とする請求項10に記載の音楽自動生成プログラム。  
 【請求項15】 前記音楽自動生成プログラムであって、前記XY座標値取得ステップは、前記平面内における始点のX座標値SXと、始点のY座標値SYと、X方向の増分値DXと、Y方向の増分値DYとを取得し、始点を第1の座標点とし、(始点のX座標値SX+m\*DX, 始点のY座標値SY+m\*DY)を第n番目の座標点とする(m, nは自然数)ことを特徴とする請求項10乃至14のいずれかに記載の音楽自動生成プログラム。  
 【請求項16】 前記音楽自動生成プログラムであって、さらに、文字列を取得するステップと、前記文字列を構成する各文字を文字コードに変換することによって前記文字列を文字コード列に変換するステップと、前記文字コード列を構成する各英数字をビット列に変換することによって前記文字コード列から第1のビット列を生成するステップと、前記第1のビット列から複数のビットを抜き出して得られる複数の第2のビット列のそれぞれを数値に変換することによって複数の数値V1, V2, V3, V4を生成するステップと、  
 30 前記複数の数値V1, V2, V3, V4から前記SX, SY, DX, DYを生成するステップと、をコンピュータに実行させることを特徴とする請求項15に記載の音楽自動生成プログラム。  
 【請求項17】 少なくとも、複素平面内における始点のX座標となる数値SXと、始点のY座標となる数値SYと、X方向の増分となる数値DXと、Y方向の増分となる数値DYとを取得するステップと、  
 $Z(0) = c$   
 $Z(n) = Z(n-1)^2 + c$   
 但し、 $c$ の実数部分は数値SX、 $c$ の虚数部分は数値SY  
 $n$ は自然数  
 で表されるZ(n)の絶対値が発散するまでの演算繰り返し回数Tを求めるステップと、  
 $Z(0) = c$   
 $Z(n) = Z(n-1)^2 + c$   
 但し、 $c$ の実数部分は(X座標値SX+m\*増分値DX)、 $c$ の虚数部分は(Y座標値SY+m\*増分値DY)  
 50 m, nは自然数

で表される  $Z(n)$  の絶対値が発散するまでの演算繰り返し回数  $T$  を求めるステップと、

0.01 以上 0.99 以下の数値を取得して、この数値を  $X(0)$  とするステップと、

$$X(n) = \lambda * X(n-1) (1 - X(n-1))$$

但し、 $\lambda$  は 4 未満の正の数

$n$  は自然数

で表される  $X(n)$  を順次計算して、各  $X(n)$  を算出するステップと、

前記各演算繰り返し回数  $T$  が奇数か否かと、前記  $X(n)$  と予め設定したしきい値との大小関係と、を判断するステップと、

前記奇数か否かの判断結果と、前記大小関係の判断結果とに基づいて、「ノートオン（音を鳴らす）」、「前の音を継続」または「休符」のいずれかを割り当てることによってリズムを生成するステップと、をコンピュータに実行させることを特徴とする音楽自動生成プログラム。

【請求項 18】 前記音楽自動生成プログラムであって、さらに、

前記数値  $V1, V2, V3, V4$  のいずれかをパラメータとして 0.01 以上 0.99 以下の数値  $X(0)$  を生成するステップと、

$$X(n) = \lambda * X(n-1) (1 - X(n-1))$$

但し、 $\lambda$  は 4 未満の正の数

$n$  は自然数

で表される  $X(n)$  を順次計算して、各  $X(n)$  を算出するステップと、

前記各演算繰り返し回数  $T$  が奇数か否かと、前記  $X(n)$  と予め設定したしきい値との大小関係と、を判断するステップと、

前記奇数か否かの判断結果と、前記大小関係の判断結果とに基づいて、「ノートオン（音を鳴らす）」、「前の音を継続」または「休符」のいずれかを割り当てることによってリズムを生成するステップと、をコンピュータに実行させることを特徴とする請求項 10 乃至 16 に記載の音楽自動生成プログラム。

【請求項 19】 平面内における所定数の座標点の  $X$  座標値と  $Y$  座標値を取得する  $XY$  座標値取得ステップと、前記所定数の座標点のそれぞれに対応する数値を取得する座標点対応数値取得ステップと、

前記座標点対応数値を所定の音階数で割った余りに従って音を割り当てることによって音列を生成するステップと、をコンピュータに実行させることを特徴とする音楽自動生成プログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、音楽を自動的に生成する方法、プログラムおよびそのようなプログラムを

記録したコンピュータ読み取り可能な記録媒体に関する。

【0002】

【従来の技術】 従来の音楽自動生成方法の中には、予め記憶されている和音データ等の音楽データの中からユーザ指定した音楽データを組み合わせて音楽を自動的に生成するものの他に、マンデルブロー写像を含むフラクタル図形のカオス的な性質を利用するものもある。

【0003】

【発明が解決しようとする課題】 しかし、予め記憶されている音楽データを組み合わせて音楽を自動的に生成する場合、その組み合わせ数には当然限界がある。特に、4～8 小節程度の短い音楽を自動生成する場合、例えば携帯電話や PHS 等の携帯通信端末の着信通知メロディー（いわゆる、着メロ）を自動生成する場合、似通った音楽が生成されやすいという問題があった。

【0004】 また、フラクタル図形のカオス的な性質を利用することによって、バリエーションの幅を広げることとは可能である。しかし、フラクタル図形等を利用する場合、ユーザが始点座標等を数値で指定する必要があったり、また生成される音楽がいわゆる現代音楽またはミニマルミュージックと呼ばれる聴感的につなりの良くない音楽になってしまうという問題があった。

【0005】 本発明はこのような課題を解決すべく新たなものであり、ユーザが音楽生成のためのパラメータとなる数値を指定する必要が無く、また似通った音楽が生成され難く、さらに生成される音楽が現代音楽風ではない聴感的につなりの良いものである音楽自動生成方法、音楽自動生成プログラムおよび音楽自動生成プログラムを記録したコンピュータ読み取り可能な記録媒体を提供することを目的とする。

【0006】

【課題を解決するための手段】 上記の課題を解決するための本発明の第 1 の特徴は、(1) 平面内における所定数の座標点の  $X$  座標値と  $Y$  座標値を取得する  $XY$  座標値取得ステップと、(2) 前記所定数の座標点のそれぞれに対応する数値を取得する座標点対応数値取得ステップと、(3) 前記座標点対応数値を所定の音階数で割った余りに従って音を割り当てることによって音列を生成するステップと、を少なくとも含むことにある。

【0007】 ここで「所定数の座標点」における「所定数」とは、音列を構成する音符の総数を言う。例えば、音列を 128 個の 16 分音符から構成する場合、「所定数」は「128」となり、音列を 64 個の 8 分音符から構成する場合、「所定数」は「64」となる。

【0008】 また、「所定の音階数」とは、使用する音階数を言う。1 オクターブは通常 8 音または 5 音（ペンタトニックスケール）なので、1 オクターブを使用する場合、所定の音階数は「8」または「5」となり、3 オクターブを使用する場合、「所定の音階数」は「24」

または「15」となる。

【0009】また、「音階数で割った余りに従って音を割り当てる」とは、例えば1オクターブを使用する場合、座標点对応数値を音階数「8」または「5」で割った余りが「1」なら「ド」を、余りが「2」なら「レ」を、余りが「3」なら「ミ」を割り当てることを言う。

【0010】なお、全音階スケールの場合は1オクターブの音階数が「7」となり、半音階スケールの場合は「12」となる。

【0011】本発明の第2の特徴は、前記音楽自動生成方法であって、さらに、前記平面内の各座標点のそれぞれに予め数値を対応づけておくステップを含むことにある。

【0012】本発明の第3の特徴は、前記音楽自動生成方法であって、前記座標点对応数値取得ステップは、画像を読み込み、画像を構成する各点の色の明るさを決定する数値を前記座標点のそれぞれに対応する数値として取得することにある。

【0013】「色の明るさを決定する数値」には、例えばRGBデータが含まれる。RGBデータとは、256色モードの場合は8ビットデータ、65536色モードの場合は16ビットデータ、1677万色モードの場合は24ビットデータとなる。

【0014】本発明の第3の特徴は、前記音楽自動生成方法であって、前記座標点对応数値取得ステップは、前記座標点の座標値とフラクタル関数とを用いて、各座標点に対応する数値を取得することにある。

【0015】「フラクタル関数」とは、自己相似性を有するフラクタル図形を描画するために用いられる関数を言う。自己相似性とは、図形の一部を取り出したとき、それが図形全体の縮小図になっているという性質を言う。

【0016】「フラクタル図形」には、マンデルブロー写像、コッホ曲線などが含まれる。

【0017】本発明の第4の特徴は、前記音楽自動生成方法であって、前記平面が複素平面であり、前記座標点对応数値取得ステップは、

$$Z(0) = c$$

$$Z(n) = Z(n-1)^2 + c$$

但し、cの実数部分は前記X座標値、cの虚数部分は前記Y座標値

nは自然数

で表されるZ(n)の絶対値が発散するまでの演算繰り返し回数Tを求めて、該演算繰り返し回数Tを前記座標点对応数値とすることにある。

【0018】Z(n)の「絶対値」とは、Z(n)の実数部分の2乗と虚数部分の2乗との和の平方根をいう。

「発散」とは、Z(n)が所定値を超えることをいう。

【0019】Z(n)の絶対値が発散するまでの演算繰り返し回数Tとは、

$$Z(1) = Z(0)^2 + c$$

$$Z(2) = Z(1)^2 + c$$

:

というように演算を繰り返して順次Z(n)を求め、このようにして求められたZ(n)の絶対値が所定値を超えたときの数値nをいう。

【0020】このようにして得られる音列は、ランダムではないが、音程変化に富んでいる。

【0021】本発明の第5の特徴は、前記音楽自動生成方法であって、前記XY座標値取得ステップは、(1)前記平面内における始点のX座標値SXと、始点のY座標値SYと、X方向の増分値DXと、Y方向の増分値DYとを取得し、(2)始点を第1の座標点とし、(3)(始点のX座標値SX+m\*増分値DX、始点のY座標値SY+m\*増分値DY)を第n番目の座標点とする(m、nは自然数)ことにある。

【0022】始点の座標値に、m\*増分値を加えることによって、始点からm\*増分値だけ離れた所定数の座標点を取得することができる。

【0023】本発明の第6の特徴は、前記音楽自動生成方法であって、さらに、(1)文字列を取得するステップと、(2)前記文字列を構成する各文字を文字コードに変換することによって前記文字列を文字コード列に変換するステップと、(3)前記文字コード列を構成する各英数字をビット列に変換することによって前記文字コード列から第1のビット列を生成するステップと、

(4)前記第1のビット列から複数のビットを抜き出して得られる複数の第2のビット列のそれぞれを数値に変換することによって複数の数値V1、V2、V3、V4を生成するステップと、(5)前記複数の数値V1、V2、V3、V4から前記SX、SY、DX、DYを生成するステップと、を含むことにある。

【0024】数値(V1、V2、V3、V4)から数値(SX、SY、DX、DY)を生成する計算式としては、例えば、

$$SX = -2.0 + V1 / 0x10000 * 4.0$$

$$SY = -2.0 + V1 / 0x10000 * 4.0$$

$$DX = 1 / (V3 + 10)$$

$$DY = 1 / (V4 + 10)$$

などが挙げられる。

【0025】このような第6の特徴によって任意の文字列から、第5の特徴を用いて音楽を自動的に生成するのに適した数値(SX、SY、DX、DY)を得ることができる。増分値(DX、DY)があまりに小さすぎると、浮動小数点の有効桁数以下となり実質的に「0」として処理されてしまう。また、「0」として処理されなくても増分値(DX、DY)が小さいと、取得される座標点の間隔が狭すぎて、各座標点に対応する数値が近似してしまふ。このため、得られる音が近似してしまひ、同じ音が続く単調な音楽が生成されてしまふ。しかし、

第6の特徴によって得られた数値(SX, SY, DX, DY)を用いて、所定数の座標点を取得すると、取得された座標点に対応する数値から生成される音楽が単調ではない、動きのある楽曲となる。

【0026】本発明の第7の特徴は、(1)複素平面内における始点のX座標値SXと、始点のY座標値SYと、X方向の増分値DXと、Y方向の増分値DYとを取得するステップと、

(2)

$$Z(0) = c$$

$$Z(n) = Z(n-1)^2 + c$$

但し、cの実数部分は始点のX座標値SX、cの虚数部分は始点のY座標値SY

nは自然数

で表されるZ(n)の絶対値が発散するまでの演算繰り返し回数Tを求めるステップと、

(3)

$$Z(0) = c$$

$$Z(n) = Z(n-1)^2 + c$$

但し、cの実数部分は(始点のX座標値SX+m\*増分値DX)、cの虚数部分は(始点のY座標値SY+m\*増分値DY)

m、nは自然数

で表されるZ(n)の絶対値が発散するまでの演算繰り返し回数Tを求めるステップと、

(4) 0.01以上0.99以下の数値を取得して、この数値をX(0)とするステップと、

(5)

$$X(n) = \lambda * X(n-1) * (1 - X(n-1))$$

但し、λは4未満の正の数

nは自然数

で表されるX(n)を順次計算して、各X(n)を算出するステップと、

(6) 前記各演算繰り返し回数Tが奇数か否かと、前記X(n)と予め設定したしきい値との大小関係と、を判断するステップと、

(7) 前記奇数か否かの判断結果と、前記大小関係の判断結果とに基づいて、「ノートオン(音を鳴らす)」、「前の音を継続」または「休符」のいずれかを割り当てることによってリズムを生成するステップと、を少なくとも含むことにある。

【0027】このようにして生成されたリズムは、カオス的ではあるがランダムではないという振る舞いを持つ。これにより、十分に変化に富んだリズムが生成される。「ランダムではない」とは、同一の初期値からスタートすれば同一の結果が得られるという算術乱数の性質を有することをいい、同一の初期値からスタートしても同一の結果が得られない物理乱数とは異なるという意味である。

【0028】算術乱数には、ロジスティック関数、レー

マー法、平方採中法、デジタルカオス法、メルセンヌ・ツイスター法、ハッシュ関数、テナント関数等が含まれる。

【0029】本発明の第8の特徴は、前記音楽自動生成方法であって、さらに、(1)前記数値V1, V2, V3, V4のいずれかをこれら数値が取りうる最大値で割った値に0.01を加えて数値X(0)を生成するステップと、

(2)

$$10 \quad X(n) = \lambda * X(n-1) * (1 - X(n-1))$$

但し、λは4未満の正の数

nは自然数

で表されるX(n)を順次計算して、各X(n)を算出するステップと、

(3) 前記各演算繰り返し回数Tが奇数か否かと、前記X(n)と予め設定したしきい値との大小関係と、を判断するステップと、

(4) 前記奇数か否かの判断結果と、前記大小関係の判断結果とに基づいて、「ノートオン(音を鳴らす)」、「前の音を継続」または「休符」のいずれかを割り当てることによってリズムを生成するステップと、を少なくとも含むことにある。

【0030】このようにして生成された音楽は、音およびリズムともに変化に富んだものとなる。

【0031】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態について説明する。

【0032】図1は、本発明の実施の形態に係る音楽生成方法の処理の概要を示すフローチャートである。図1に示すように、本発明の実施の形態に係る音楽生成方法は、文字列から数値を生成し、さらに音列とリズムを生成し、最後にリズムを音列にあてはめて音楽を生成する。以下、各処理について具体的に説明する。

【0033】(A)文字列から数値の生成

図2は、文字列から数値を生成する処理の内容を示すフローチャートである。図2に示すように、まず文字列を文字コード列に変換し(ステップS110)、次に文字コード列からビット列を生成し(ステップS120)、そしてビット列から数値を生成する(ステップS130)。

【0034】(A-1)文字列を文字コード列へ変換する(ステップS110)

文字列を文字コード列へ変換するためには、文字と文字コードとの対応関係を示す文字コードテーブルが必要である。そして、文字列を各文字に分け、各文字について文字コードテーブルを参照して対応する文字コードを調べる処理を繰り返すことによって、文字列を文字コード列へ変換する。

【0035】図3は、文字列と文字コード列の対応関係を示す図である。図3は、文字列が「プレイバック Par

t2」の例であり、各文字をシフトJISコードまたはASCIIコードに変換する。つまり、「ブ」を「8376」に、「レ」を「838C」に、「イ」を「8343」に、「パ」を「836F」に、「ッ」を「8362」に、「ク」を「834E」に、「」（半角スペース）を「20」に、「P」を「50」に、「a」を「61」に、「r」を「72」に、「t」を「74」に、「2」を「32」にそれぞれ変換する。このようにして文字列「プレイバック Part2」から、コード列「8376 838C 8343 836F 8362 834E 20 50 61 72 74 32」が得られる。

【0036】(A-2)文字コード列からビット列を生成する(ステップS120)

文字コードを構成する各英数字(0~9とA~F)から4ビットのビット列を生成する。つまり、「0」から「0000」を、「1」から「0001」を、「2」から「0010」を、・・・「F」から「1111」を生成する。

【0037】この際、文字コード列中に出現する英数字には偏りがあるので、この偏りを少なくする処理をすることが好ましい。

【0038】例えば、シフトJISコードの最上位の英数字「8」、「9」または「E」を取り除き、詰めてしまう。

【0039】図4は、文字コード列からビット列を生成する際に、英数字の偏りを少なくする処理の一例を示す図である。図4に示すように、「8376」を「376」とし、「838C」を「38C」とし、「8343」を「343」とし、「836F」を「36F」とし、「8362」を「362」とし、「834E」を「34E」とすることによって、「8」等の出現頻度を少なくする。

【0040】その後、各英数字列をビット列に変換する。つまり、「376」を「001101110110」とし、「38C」を「001110001100」とし、「343」を「001101000011」とし、「36F」を「001101101111」とし、「362」を「001101100010」とし、「34E」を「001101001110」とする。

【0041】また、ASCIIコードの各数値を8ビットのビット列に変換した場合の最上位ビットは、常に「0」であるから、これを取り除いて、詰めることによって、「0」の出現頻度を少なくする。つまり、「20」を「00100000」を経て「0100000」とし、「50」を「01010000」を経て「1010000」とし、「61」を「01100001」を経て「1100001」とし、「72」を「01110010」を経て「1110010」とし、「74」を「01110100」を経て「1110100」とし、「32」を「00110010」を経て「0110010」とする。このようにして、文字コード列「8376 838C 8343 836F 8362 834E 2050 61 72 74 32」から、ビット列「0011011101100011100011000011010000110011011011100110110001000110100111001000001010000110000111100101101000110010」が得られる。

【0042】(A-3)ビット列から数値を生成する

(ステップS130)

次に、ビット列から4つの数値(V1~V4)を生成する。コード列から生成されたビット列が64ビット以上である場合は、最上位ビットまたは最下位ビットから、16ビット毎に分割して、16ビットのビット列を4つ生成しても良い。

【0043】また、コード列から生成されたビット列が64ビット未満である場合は、「0」を補完して、64ビットのビット列とした後に、16ビット毎に分割しても良い。

【0044】また、64ビットのビット列の最上位ビットまたは最下位ビットから、4ビットおきに16個のビットを抜き出して、16ビットのビット列を4つ生成しても良い。

【0045】図5に、64ビットのビット列から4ビットおきに16個のビットを抜き出して、16ビットのビット列を生成して、数値Vを取得する処理の一例を示す。

【0046】図5(a)に示すように、ビット列「001101101100011100011000011010000110011011011110011011000100011」から、1番目の「0」、5番目の「0」、9番目の「0」、13番目の「0」、17番目の「1」、21番目の「1」、25番目の「0」、29番目の「0」、33番目の「0」、37番目の「0」、41番目の「0」、45番目の「1」、49番目の「0」、53番目の「0」、57番目の「0」、61番目の「0」を抜き出して、ビット列「0000110000010000」を生成する。そして、得られたビット列から数値V1を得る。

【0047】同様に、図5(b)に示すように、ビット列「001101110110001110001100001101000011001101101110011011000100011」から、2番目の「0」、6番目の「1」、10番目の「1」、14番目の「0」、18番目の「0」、22番目の「1」、26番目の「0」、30番目の「1」、34番目の「0」、38番目の「0」、42番目の「1」、46番目の「1」、50番目の「0」、54番目の「1」、58番目の「0」、62番目の「0」を抜き出して、ビット列「0110010100110100」を生成する。そして、得られたビット列から数値V2を得る。

【0048】4ビットおきに抜き出して生成する方が、文字列中のほんの1部でも変化すると、全体が変化するため、好ましい。

【0049】このようにして、4つの数値(以下、V1、V2、V3、V4とする。)を生成する。

【0050】(B)数値から音列を生成する(ステップS200)

図6は、数値から音列を生成する処理の流れを示すフローチャートである。図6に示すように、まず数値(V1、V2、V3、V4)から数値(SX、SY、DX、DY)を得る(ステップS210)。SXとSYはそれぞれ始点のX座標とY座標を表し、DXとDYはそれぞれ

10

20

30

40

50

れX方向の増分とY方向の増分を表す。

【0051】次に、数値(SX、SY、DX、DY)から128個の繰り返し回数Tを求める(ステップS220)。まず始点(SX、SY)について繰り返し回数Tを求め、次に2番目の点(SX+DX、SY+DY)について繰り返し回数Tを求める。以下同様にして、128番目の点(SX+127\*DX、SY+127\*DY)について繰り返し回数を求める。

【0052】最後に、128個の繰り返し回数Tから128個の音を求める(ステップS230)。各ステップの具体的内容について説明する。

【0053】(B-1)数値(V1、V2、V3、V4)から数値(SX、SY、DX、DY)を得る(ステップS210)

例えば、下記式を用いて4つの数値(V1、V2、V3、V4)から4つの数値(SX、SY、DX、DY)を得る。

【0054】

$$SX = -2.0 + V1 / 0x10000 \times 4.0$$

$$SY = -2.0 + V2 / 0x10000 \times 4.0$$

$$DX = 1 / (V3 + 10)$$

$$DY = 1 / (V4 + 10)$$

始点(SX、SY)は、(-2、-2)、(-2、2)、(2、2)および(2、-2)で囲まれる四角形の中に入る。つまり $2 \geq |X|$ 、 $2 \geq |Y|$ である。始点をこの範囲に入るものとしたのは、この範囲外では少ない繰り返し回数でZ(n)の絶対値が発散に至ってしまうためである。例えば、 $|X| \geq 3$ 、 $|Y| \geq 3$ だとZ(5)程度で発散してしまう。つまり繰り返し回数が常に「5」になってしまい、生成される音楽が単調にな

ってしまう。  
【0055】(B-2)数値(SX、SY、DX、DY)から128個の繰り返し回数Tを求める(ステップS220)

マンデルブロー関数等のフラクタル関数を用いて128個の座標点について繰り返し回数Tを求める。

【0056】具体的には、始点(SX、SY)、2番目の点(SX+DX、SY+DY)、3番目の点(SX+2\*DX、SY+2\*DY)、・・・128番目の点(SX+127\*DX、SY+127\*DY)の各点について、マンデルブロー関数の値|Z|が所定値を超えるまでの漸化式の繰り返し回数Tを求める。

【0057】図7に、マンデルブロー関数を用いて始点座標(SX、SY)について繰り返し回数Tを求める処理を示す。

【0058】まず、始点(SX、SY)についてマンデルブロー関数の値が発散するまでの演算繰り返し回数を求める(ステップS310)。

【0059】一般的に、マンデルブロー関数とは、 $Z(n) = Z(n-1)^2 + c$  (nは自然数)

で表される漸化式をいう。cは、 $c = a + ib$  (a、bは実数であり、aを実数部分、bを虚数部分という。iは-1の平方根である。)と表される。また、 $(a^2 + b^2)$ の平方根をcの絶対値といい、|c|と表す。

【0060】本実施形態においては、始点のX座標(SX)を複素数cの実数部分とし、始点のY座標(SY)を虚数部分とし、また|Z|が2を超えたら「発散」とみなす。

【0061】具体的には、図7に示すように、

$$Z(0) = SX + iSY$$

$$Z(1) = Z(0)^2 + Z(0)$$

$$Z(2) = Z(1)^2 + Z(0)$$

:

:

$$Z(n) = Z(n-1)^2 + Z(0)$$

という様にZ(n)を順次求め、かつ各Z(n)について|Z(n)|が2を超えたか否かを調べる。|Z(n)|が2を超えた時のnを、発散するまでの演算繰り返し回数Tとする。なお、計算を400回繰り返しても発散しない場合は、繰り返し回数T=400とする。

【0062】このようにして(SX、SY)についての繰り返し回数Tを求める。

【0063】次に、2番目の点(SX+DX、SY+DY)についての繰り返し回数Tを求める。

【0064】図8(a)に、始点(SX、SY)に増分を加えた座標(SX+DX、SY+DY)について、発散までの演算繰り返し回数を求める計算式を示す。

【0065】図8(a)に示すように、

$$Z(0) = (SX+DX) + i(SY+DY)$$

$$Z(1) = Z(0)^2 + Z(0)$$

$$Z(2) = Z(1)^2 + Z(0)$$

:

:

$$Z(n) = Z(n-1)^2 + Z(0)$$

という様にZ(n)を順次求め、かつ各Z(n)について|Z(n)|が2を超えたか否かを調べる。|Z(n)|が2を超えた時のnを、発散するまでの演算繰り返し回数Tとする。始点(SX、SY)と同様に、計算を400回繰り返しても発散しない場合は、繰り返し回数T=400とする。

【0066】ここまでで始点座標(SX、SY)と2番目の座標(SX+DX、SY+DY)という2つの座標のそれぞれについて、演算繰り返し回数Tが求まる。

【0067】図8(b)に示すように、さらに同様にして3番目の点(SX+2\*DX、SY+2\*DY)から128番目の点(SX+127\*DX、SY+127\*DY)までについて繰り返し回数を求めることによって、128個の繰り返し回数Tを得る。なお、128番目まで求めるのは、1トラックを128個の16分音符



で構成するとしたためである。

【0068】1トラックを構成する音の数を増減した場合は、必要となる繰り返し回数 $T$ の個数も増減する。

【0069】なお、 $n$ 番目の座標( $X_n, Y_n$ )についての繰り返し回数 $T(n)$ と、 $n$ 番目の座標に増分( $DX, DY$ )を加えた座標( $X_n+DX, Y_n+DY$ )についての繰り返し回数 $T$ とが同じ値になってしまう場合は、 $n$ 番目の座標に増分( $DX, DY$ )の2倍を加えた座標( $X_n+2*DX, Y_n+2*DY$ )について求めた繰り返し回数 $T$ を $T(n+1)$ とすることが好ましい。

【0070】また、 $n$ 番目の座標( $X_n, Y_n$ )についての繰り返し回数 $T(n)$ と、 $n$ 番目の座標に増分( $DX, DY$ )の2倍を加えた座標( $X_n+2*DX, Y_n+2*DY$ )についての繰り返し回数 $T$ とが同じ値になってしまう場合は、異なる繰り返し回数 $T$ が得られるまで、増分( $DX, DY$ )の加算を繰り返すことが好ましい。つまり、異なる繰り返し回数 $T$ が得られるまで、( $X_n+2*DX, Y_n+2*DY$ )、( $X_n+3*DX, Y_n+3*DY$ )、( $X_n+4*DX, Y_n+4*DY$ )、 $\dots$ のそれぞれについて繰り返し回数 $T$ を求める。

【0071】さらに、 $n$ 番目の座標( $X_n, Y_n$ )についての繰り返し回数 $T(n)$ と、 $n$ 番目の座標に増分( $DX, DY$ )の2倍を加えた座標( $X_n+2*DX, Y_n+2*DY$ )についての繰り返し回数 $T$ とが同じ値になってしまった場合に、異なる繰り返し回数 $T$ が得られるまで、増分( $DX, DY$ )の2倍の加算を繰り返すことがより好ましい。つまり、異なる繰り返し回数 $T$ が得られるまで、( $X_n+2*DX, Y_n+2*DY$ )、( $X_n+4*DX, Y_n+4*DY$ )、( $X_n+6*DX, Y_n+6*DY$ )、 $\dots$ のそれぞれについて繰り返し回数 $T$ を求める。増分を大きくすることによって、異なる繰り返し回数 $T$ を得るまでに必要な計算の総数を減少させることが可能となる。

【0072】また、増分( $DX, DY$ )を加算することによって得られた新たな座標が、(2.0,2.0)と(2.0,-2.0)と(-2.0,2.0)と(-2.0,-2.0)の4点によって囲まれる四角形の外に出てしまう場合は、図9に示すように、ラップ・アラウンド(上辺からはみ出したら下辺へ移動し、右辺からはみ出したら左辺へ移動する)という処理をすることが好ましい。

【0073】(多声化)なお、単音での生成でも音楽として成立するが、「聞くに耐える音楽」とするには複数の音の方が音楽的幅が広がるため好ましい。

【0074】「聞くに耐える音楽」とは、十分な音程変化があり、さらにリズム的にも十分複雑な旋律を持った音楽のことである。さらに、複数の旋律が短2度等の不協和音にならない限り、できるだけ各旋律同士がうまく絡み合わさることによって、主旋律を強調したり、主旋

律以外の旋律の動きがリズムに聞こえるとより一層音楽的といえる。

【0075】例えば、1番目のトラックの終点( $SX+127*DX, SY+127*DY$ )から所定値だけ離れた点を2番目のトラックの始点として、上記と同じ処理を繰り返すことによって、2番目のトラック用の128個の繰り返し回数を求める。同様の処理を繰り返して、3番目から8番目のトラック用の128個の繰り返し回数を求める。

【0076】 $m$ 番目のトラックの終点と( $m+1$ )番目のトラックの始点の座標が近接していると、 $m$ 番目のトラックの数列パターンと( $m+1$ )番目のトラックの数列パターンが似てしまい、その結果、 $m$ 番目のトラックの音列と( $m+1$ )番目のトラックの音列のパターンが似てしまう。ここで、 $m$ は7以下の自然数である。

【0077】一方、 $m$ 番目のトラックの終点と( $m+1$ )番目のトラックの始点の座標が離れ過ぎていると、 $m$ 番目のトラックの数列パターンと( $m+1$ )番目のトラックの数列パターンの関連性が希薄になり、その結果、 $m$ 番目のトラックの音列と( $m+1$ )番目のトラックの音列のパターンの関連性が希薄になってしまう。

【0078】そこで、 $m$ 番目のトラックの始点と( $m+1$ )番目のトラックの始点との間隔を以下のようにして決定することが好ましい。なお、以下の処理は $X$ 座標および $Y$ 座標のそれぞれについて行うことが好ましい。

【0079】( $m$ 番目のトラックの終点座標- $m$ 番目のトラックの始点座標)を10で割って得られた値が0.01より小さい場合は、得られた値をそのまま、 $m$ 番目のトラックの終点と( $m+1$ )番目のトラックの始点との間隔とする。

【0080】10で割って得られた値が0.01より大きい場合は、0.01より小さくなるまで7で割る。

【0081】隣接するトラックから生成される音列は、似ているが少し違うことが好ましい。隣接するトラックから生成される音列が、あまりにかけ離れているとまとまりのない音楽になってしまう。隣接するトラックの始点が接近しすぎると、両トラックから生成される音列が同じになってしまい好ましくない。

【0082】( $m$ 番目のトラックの終点座標- $m$ 番目のトラックの始点座標)を「10」で割る、とした。しかし「10」に限らずに、「5」から「50」の間の数値で割っても「似ているが少し違う」音列を生成することは可能である。ただし、短時間で計算可能という理由で「10」が好ましい。

【0083】また、10で割って得られた値が0.01より大きい場合は、0.01より小さくなるまで「7」で割る、とした。しかし「7」に限らず、「10」以外で「10」に近い他の数値(例えば、「5~15」)で割る、としても「似ているが少し違う」音列を得ることができる。

【0084】全部で8トラック分の数列が得られたら全てのトラックを比較し、「あるトラックの第1小節を構成する音列に対応する数列」と「他のトラックの第1小節を構成する音列に対応する数列」が完全に一致する場合は、以下の操作を行った上で再計算することが好ましい。

【0085】例えば、m番目のトラックの終点を、増分に関係なく(0,0)とする。そして、m番目のトラックの終点と(m+1)番目のトラックの始点との間隔を(m番目のトラックの終点-m番目のトラックの始点)  
\*2/10とする。ただし、(m番目のトラックの終点-m番目のトラックの始点)\*2/10が、0.001より小さい場合に限る。

【0086】(B-3)各繰り返し回数Tから音を生成する(ステップS230)

各繰り返し回数を音へ変換する場合、使用する音階を予め定義し、かつ各音に通し番号を付与しておく。そして、各繰り返し回数を音階の総個数で割り、得られた剰余と等しい番号が付与されている音を、各繰り返し回数の音とする。

【0087】図10に、「ドレミファソラシ」の各音を、繰り返し回数Tに割り当てる例を示す。図10に示すように、繰り返し回数Tを7で割った余りが「0」なら「ド」、「1」なら「レ」、「2」なら「ミ」、「3」なら「ファ」、「4」なら「ソ」、「5」なら「ラ」、「6」なら「シ」というように割り当てる。

【0088】このように割り当てた場合、1番目の繰り返し回数T(0)が「70」なら1番目の音は「ド」、2番目の繰り返し回数T(1)が「71」なら2番目の音は「レ」、3番目の繰り返し回数T(2)が「72」なら3番目の音は「ミ」、・・・、100番目の繰り返し回数T(99)が「350」なら100番目の音は「ド」、・・・128番目の繰り返し回数T(127)が「286」なら128番目の音は「シ」となる。

【0089】(C)リズムの生成

図11は、数値V1と繰り返し回数T(0)~T(127)を用いてリズムを生成する処理の流れを示すフローチャートである。数値V1の代わりに、数値V2、V3、V4のいずれかを用いても良い。

【0090】図11に示すように、まず数値V1から128個の数値X(0)~X(127)を生成し(ステップS310)、その後これら数値X(0)~X(127)と繰り返し回数T(0)~T(127)からノートオン、前音の継続または休符のいずれかを指示する128個の命令を生成する(ステップS320)。以下、具体的に説明する。

【0091】(C-1)数値V1から128個の数値X(0)~X(127)を生成する(ステップS310)  
数列X(n)において(nは自然数)、X(n)とX(n+1)との関係が

$$X(n+1) = \lambda * X(n) * (1 - X(n)) \quad 0 \leq \lambda \leq 4$$

で与えられているとき、このような数列をロジスティックモデルという。この数列は不規則な値をとる。すなわちカオス的な振る舞いをする。

【0092】例えば、 $\lambda = 3.98$ とし、また、X(n)の初期値X(0)は、前記V1を再利用して、 $X(0) = 0.01 + 0.98 * V1 / 65535$ とする。この式において、V1=0ならX(0)=0.01となり、V1=65535(16ビットで表現できる最大値)ならX(0)=0.99となる。

【0093】図12に、ロジスティック関数を用いてX(n)を順次計算する例を示す。図12(a)に示すように、まず、

$$X(0) = 0.01 + 0.98 * V1 / 65535$$

からX(0)を求める。次に、このX(0)を

$$X(1) = \lambda * X(0) * (1 - X(0))$$

に代入してX(1)を求める。以下同様にして、

$$X(2) = \lambda * X(1) * (1 - X(1))$$

:

:

$$X(126) = \lambda * X(125) * (1 - X(125))$$

$$X(127) = \lambda * X(126) * (1 - X(126))$$

と順次計算を繰り返して、X(0)からX(127)までの数値を求める。

【0094】仮に、 $\lambda = 3.98$ 、V1=100とすると、

$$X(0) = 0.01 + 0.98 * 100 / 65535 = 0.01149538414$$

$$X(1) = 3.98 * 0.01149538414 * (1 - 0.01149538414) = 0.045225$$

$$69632$$

$$X(2) = 3.98 * 0.04522569632 * (1 - 0.04522569632) = 0.171857$$

$$72419$$

$$X(3) = 3.98 * 0.17185772419 * (1 - 0.17185772419) = 0.566444$$

$$13436$$

:

となる。

【0095】 $\lambda$ を一定値とし、一方V1を0以上65535以下の整数とした場合、X(0)の取りうる値は65536通りに限定される。このため、各X(0)から順次計算して得られるX(127)までの数列も65536通りに限定されてしまう。

【0096】数列が65536通りに限定されてしまうと、同じリズムが生成される確率が高くなる。同じリズムが生成される確率を下げるためには、つまり異なるリズムが生成される確率を上げるためには、各X(0)からN回計算を繰り返してX(N)を求め、このX(N)からさらに127回計算を繰り返してX(N+127)までの数値を求め、これらX(N)からX(N+127)までの128個の数値を用いてリズムを生成することが好ましい。

19

【0097】図12(b)に、 $X(0)$ から $N$ 回計算を繰り返して $X(N)$ を求め、この $X(N)$ からさらに127回計算を繰り返して $X(N)$ から $X(N+127)$ までの数値を求める例を示す。

【0098】図12(b)に示すように、

$$X(0)=0.01+0.98*V1/65535$$

$$X(1)=\lambda *X(0)*(1-X(0))$$

$$X(2)=\lambda *X(1)*(1-X(1))$$

:

$$X(N)=\lambda *X(N-1)*(1-X(N-1))$$

という計算を繰り返して、 $X(N)$ を求める。そして、さらに

$$X(N+1)=\lambda *X(N)*(1-X(N))$$

$$X(N+2)=\lambda *X(N+1)*(1-X(N+1))$$

:

$$X(N+126)=\lambda *X(N+125)*(1-X(N+125))$$

$$X(N+127)=\lambda *X(N+126)*(1-X(N+126))$$

という計算を繰り返して、 $X(N)$ から $X(N+127)$ までの各数値を求める。このようにして、1トラック分の128個の $X(n)$ が求まる。

【0099】上記数値 $N$ は、例えば入力された文字列の下位7ビット(ASCII/全角の区別無し)を全て加算した値とする。つまり、入力された文字列が“abc”である場合、各文字の文字コードは“65, 66, 67”である。これら各文字コードをビット列には変換し、下位7ビットを全て加算した値は「198」となる。

【0100】仮に、 $V1=100$ とすると、

$$X(0)=0.01+0.98*100/65535=0.01149538414$$

$$X(1)=3.98*0.01149538414*(1-0.01149538414)=0.04522569632$$

$$X(2)=3.98*0.04522569632*(1-0.04522569632)=0.1718572419$$

$$X(3)=3.98*0.1718572419*(1-0.1718572419)=0.56644413436$$

:

という計算を198回繰り返して、 $X(198)$ を求める。そして、さらに計算を127回繰り返すことによって、 $X(198)$ から $X(198+127)$ までの各 $X(n)$ の値を求める。

【0101】(C-2)128個の繰り返し回数 $T$ と128個の数値 $X$ とからリズムを生成する(ステップS320)

マンデルブロー関数を用いて得られた128個の繰り返し回数 $T(0) \sim T(127)$ と、ロジスティック関数を用いて得られた128個の数値 $X(0) \sim X(127)$ とに基づいて、「ノートオン」、「継続」または「休符」のいずれかを決定する。

【0102】例えば、

1) 繰り返し回数 $T(n)$ が奇数で、 $X(n)$ がしきい値より大きければ「ノートオン(音を鳴らす)」とし、

20

2) 繰り返し回数 $T(n)$ が奇数で、 $X(n)$ がしきい値以下である場合は前の音を「継続」とする。ただし、前に音が無ければ「ノートオン」とし、

3) 繰り返し回数 $T(n)$ が偶数で、 $X(n)$ がしきい値より大きければ「ノートオン」とし、

4) 繰り返し回数 $T(n)$ が偶数で、 $X(n)$ がしきい値以下である場合は「休符」とする。

【0103】図13は、繰り返し回数 $T$ と数値 $X$ に基づいてリズムを生成する処理の流れを示すフローチャートである。

【0104】図13に示すように、まず繰り返し回数 $T(n)$ が奇数か偶数かを判断する(ステップS321)。

【0105】繰り返し回数 $T(n)$ が奇数の場合、 $X(n)$ がしきい値より大きいか小さいかを判断する(ステップS322)。 $X(n)$ がしきい値より大きい場合は、ノートオンとする。 $X(n)$ がしきい値より小さい場合は、さらに前に音があるか/無いかを判断する(ステップS323)。前に音がある場合は前の音を継続し、前に音がない場合はノートオンとする。

【0106】繰り返し回数 $T(n)$ が偶数の場合も、 $X(n)$ がしきい値より大きいか小さいかを判断する(ステップS325)。 $X(n)$ がしきい値より大きい場合は、さらに前に音があるか/無いかを判断する(ステップS326)。前に音がある場合は前の音を継続し、前に音がない場合はノートオンとする。 $X(n)$ がしきい値より小さい場合は、休符とする。

【0107】(D)リズムを音列に当てはめて音楽を生成する

最後に、既に生成したリズムを、既に生成した音列に当てはめて、音楽を作成する。

【0108】図14に、リズムを音列に当てはめて音楽を生成する処理の一例を示す。図14に示す例では、「ノートオン」を「0」、「前の音の継続」を「1」、「休符」を「2」と表すとした場合に、生成されたリズムが「0010200201022101121012010」で表されたとする。また、「C」が「ド」、「D」が「レ」、「E」が「ミ」、「F」が「ファ」、「G」が「ソ」、「A」が「ラ」、「B」が「シ」を表すとした場合に、生成された音列が「CBFGECDBFCFDGCDFAADCBDEFG」で表されたとする。

【0109】つまり、リズムは、

0010200201022101121012010  
と表記され、音列は、  
CBFGECDBFCFDGCDFAADCBDEFG  
と表記される。

【0110】そして、前記リズムを前記音列に当てはめると、

CB-G\_CD-F-F\_\_CD--\_DC-\_E-G

10

20

30

40

50

という音楽が得られる。

【0111】16分音符ごとにノートオン、継続、休符を割り当てるとすると、「C 16分音符」、「B 8分音符」、「G 16分音符」、「16分休符」、「C 16分音符」、「D 16分音符」、「16分休符」、「F 8分音符」、「F 16分音符」、「8分休符」、「C 16分音符」、「D付点8分音符」、「16分休符」、「D 16分音符」、「C 8分音符」、「16分休符」、「E 8分音符」、「G 16分音符」からなる音楽が得られる。

【0112】以上説明したように、本発明の実施の形態によれば、文字列から数値を生成すること、音列を生成すること、リズムを生成すること、音楽を生成することができる。

【0113】このようにして生成される音列は、同一の初期値から同一の結果が得られるという性質を有するためランダムではないが、音程変化に富んでいる。

【0114】なお、文字列から音列を生成するのではなく、既存の画像データから音列を生成することもできる。例えば、(1) 256種類の色で表現された画像上から複数の座標点を取得し、(2) 各座標点に対応する色データ(0~255の数値)を取得し、(3) 取得された色データの数値を使用する音階数(例えば、「36」)で割った余りに応じて音を割り当てるとしても良い。

【0115】このようにして生成されるリズムは、ランダムではないが、カオス的な振る舞いを持ち、十分に変化に富んでいる。

【0116】また、ロジスティック関数は、(1) 同一の初期値から同一の結果が得られる、(2) 計算が簡単であるという特徴を有するため、好ましい。

【0117】計算に時間がかかってもかまわない又は複雑な式を用いても瞬時に計算を完了できるほど高機能な計算機を使用できるという場合は、複雑な計算式を用いても良い。例えば、テント写像を表す関数から求めた同相変換式 $Y(t) = \{2/\pi / \sin^2 x(t)\}$ を用いて得られる $Y(t)$ が整数値になるように量子化を施した下記式からでも同様な結果が得られる。

$$【0118】Y(t) = [ \{2/\pi / \sin^2 x(t)\} * 2^n ]$$

なお、[]は小数点以下を切り捨てる処理を意味する。

【0119】また、各座標値から数式を用いて各座標点に対応する数値を算出するのではなく、予め各座標値に対応する数値を決めておき、かかる数値を読み出すとしても良い。

【0120】このようにして生成される音楽は、十分な音程変化があり、リズム的にも十分複雑な旋律を持つ。さらに複数の旋律が短2度等の不協和音にならない限り、できるだけ各旋律同士がうまく絡み合わさる事で、主旋律を強調したり、主旋律以外の旋律の動きがリズムに聴こえる。

【0121】上記実施形態では、V1から始点のX座標値SXを生成する式として

$$SX = -2.0 + V1 / 0x10000 * 4.0$$

使用したが、

$$V1 / 0x10000$$

の代わりに微分関数

$$F'(X) = \lim_{h \rightarrow 0} (F(X+h) - F(X)) / h$$

を利用しても良い。なお「 $h \rightarrow 0$ 」は「 $|h| = 1/n$  (nは自然数)」とする。

【0122】また、128個の座標点を取得するために始点(SX, SY)に増分(DX, DY)を順次加算した。しかし、このような1次関数を用いる方法の他に、 $Y = a * X^2$ などの2次関数を用いても良い。

【0123】さらに、範囲を円周の中に求めるようなプログラムにした場合、 $\pi$ を利用した関数を使用しても良い。またさらに、その関数を微分したものを使用しても良い。

【0124】

【発明の効果】本発明によれば、複数の座標点から数値を取得し、各数値から音を生成することができる。

【0125】また、(1) 複素平面の座標点、増分およびフラクタル関数から音列を、(2) 複素平面の座標点、増分、フラクタル関数およびロジスティック関数からリズムを、(3) そして音列とリズムを組み合わせて音楽を、生成する方法、生成するプログラムおよびそのようなプログラムを記録したコンピュータ読み取り可能な記録媒体を実現することができる。

【0126】また、文字列からこのような音列およびリズムの生成に必要な複素平面の座標点および増分を生成する方法、生成するプログラムおよびそのようなプログラムを記録したコンピュータ読み取り可能な記録媒体を実現することもできる。

【図面の簡単な説明】

【図1】本発明の実施の形態に係る音楽生成方法の処理の概要を示すフローチャートである。

【図2】文字列から数値を生成する処理の内容を示すフローチャートである。

【図3】文字列と文字コード列の対応関係を示す図である。

【図4】図3に示す文字コード列からビット列を生成する際に、英数字の偏りを少なくする処理の一例を示す図である。

【図5】64ビットのビット列から4ビットおきに16個のビットを抜き出して、16ビットのビット列を生成して、数値Vを取得する処理の一例を示す図である。

【図6】数値から音列を生成する処理の流れを示すフローチャートである。

【図7】マンデルブロー関数を用いて始点座標(SX, SY)について繰り返し回数Tを求める処理を示す図で

ある。

【図8】(a)はマンデルブロー関数を用いて2番目の点( $SX+DX$ ,  $SY+DY$ )について繰り返し回数 $T$ を求める処理、(b)はマンデルブロー関数を用いて3番目の点( $SX+2*DX$ ,  $SY+2*DY$ )および128番目の点( $SX+127*DX$ ,  $SY+127*DY$ )について繰り返し回数 $T$ を求める処理を示す図である。

【図9】ラップ・アラウンド処理を示す図である。

【図10】「ドレミファソラシ」の各音を、繰り返し回数 $T$ に割り当てる例を示す図である。

【図11】数値 $V1$ と繰り返し回数 $T(0) \sim T(127)$ を用いてリズムを生成する処理の流れを示すフローチャートである。

\*

\*【図12】(a)はロジスティック関数を用いて $X(0)$ から $X(127)$ までを、(b)はロジステック関数を用いて $X(N)$ から $X(N+127)$ までを順次計算する例を示す図である。

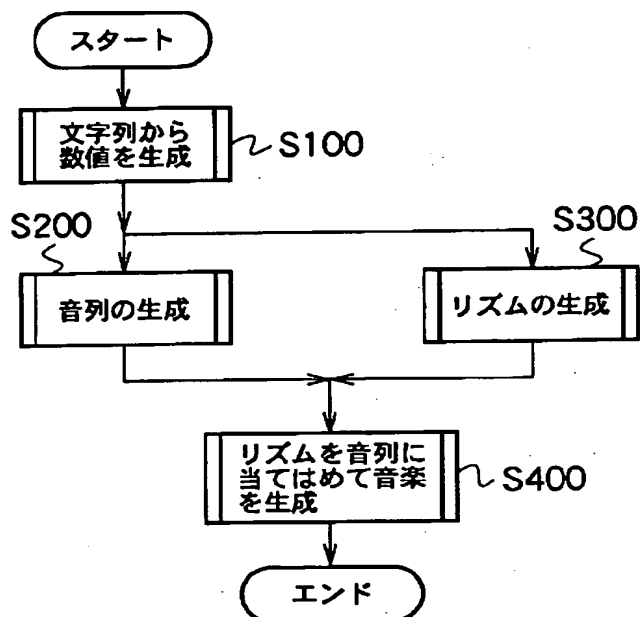
【図13】繰り返し回数 $T$ と数値 $X$ に基づいてリズムを生成する処理の流れを示すフローチャートである。

【図14】リズムを音列に当てはめて音楽を生成する処理の一例を示す図である。

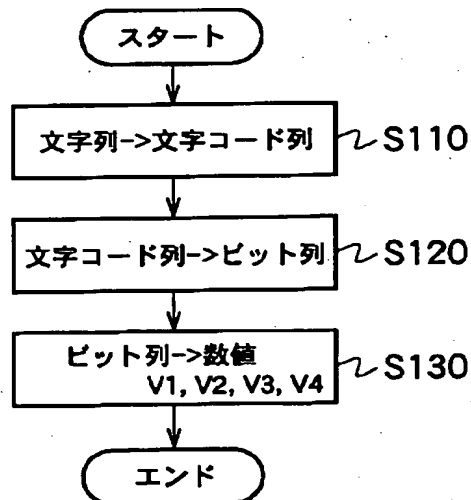
【符号の説明】

100 文字列から数値を生成するステップ  
200 音列を生成するステップ  
300 リズムを生成するステップ  
400 リズムを音列に当てはめて音楽を生成するステップ

【図1】



【図2】



【図3】

「フ」→「8376」  
「レ」→「838C」  
「イ」→「8343」  
「バ」→「836F」  
「ツ」→「8362」  
「ク」→「834E」  
「 」(半角スペース)→「20」  
「P」→「50」  
「a」→「61」  
「r」→「72」  
「t」→「74」  
「2」→「32」

文字列「プレイバック Part2」→

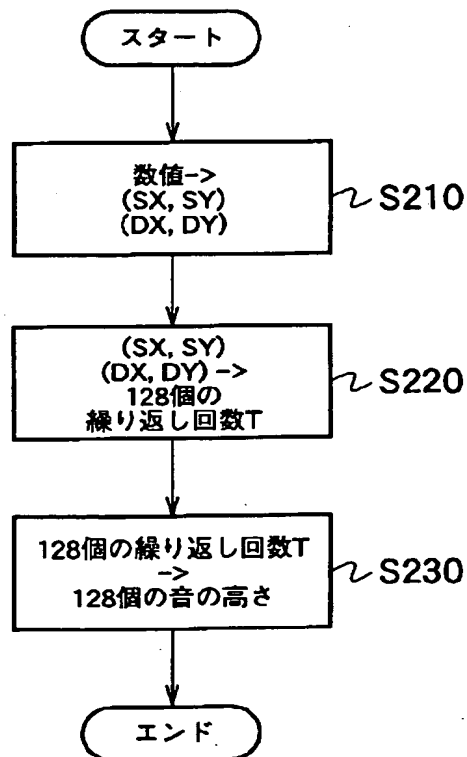
コード列「8376 838C 8343 836F 8382 834E 20 50 61 72 74 32」

【図4】

「8376」→「376」→「001101110110」  
 「838C」→「38C」→「001110001100」  
 「8343」→「343」→「001101000011」  
 「836F」→「36F」→「001101101111」  
 「8362」→「362」→「001101100010」  
 「834E」→「34E」→「001101001110」  
 「20」→「00100000」→「01000000」  
 「50」→「01010000」→「10100000」  
 「61」→「01100001」→「11000001」  
 「72」→「01110010」→「11100100」  
 「74」→「01110100」→「11110100」  
 「32」→「00110010」→「01100100」

コード列「8376 838C 8343 836F 8362 834E 20 50 6172 74 32」→  
 ビット列「0011011101100011100011000011010000110011011011100110110001000  
 1101001110010000010100001100001111001011101000110010」

【図6】



【図5】

0011 0111 0110 0011 1000 1100 0011 0100  
 0011 0011 0110 1111 0011 0110 0010 0011...

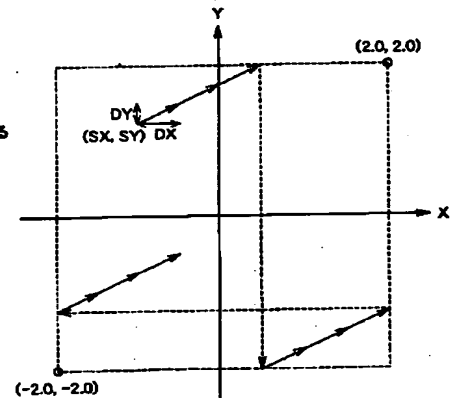
(a)  
 [0]11 [0]11 [0]10 [0]01 [1]000 [1]100 [0]01 [0]100  
 [0]01 [0]01 [0]10 [1]11 [0]01 [0]10 [0]10 [0]01...  
 →{0000110000010000}  
 → $2^{11}+2^{10}+2^4$   
 →3088  
 →V1

(b)  
 [0]11 [0]11 [0]10 [0]01 [1]00 1[1]00 [0]01 [0]00  
 [0]01 [0]01 [0]10 [1]11 [0]01 [0]10 [0]10 [0]01...  
 →{0110010100110100}  
 → $2^{14}+2^{13}+2^8+2^5+2^4+2^2$   
 →25808  
 →V2  
 :  
 :  
 :

【図7】

$SX = -0.2 + V1/10000 * 4.0$   
 $SY = -0.2 + V2/10000 * 4.0$   
 $DX = 1/(\sqrt{3}+10)$   
 $DY = 1/(\sqrt{4}+10)$   
 座標点(SX, SY)について繰り返し回数Tを求める  
 $Z(0) = SX + iSY$   
 $Z(1) = Z(0)^2 + Z(0)$   
 $Z(2) = Z(1)^2 + Z(0)$   
 $Z(3) = Z(2)^2 + Z(0)$   
 :  
 $Z(t-1) = Z(t-2)^2 + Z(0)$   
 $Z(t) = Z(t-1)^2 + Z(0)$   
 :  
 $Z(398) = Z(397)^2 + Z(0)$   
 $Z(399) = Z(398)^2 + Z(0)$

【図9】



【図10】

$T(n) \bmod 7 = 0$  なら n番目の音は「ド」  
 $T(n) \bmod 7 = 1$  なら n番目の音は「レ」  
 $T(n) \bmod 7 = 2$  なら n番目の音は「ミ」  
 $T(n) \bmod 7 = 3$  なら n番目の音は「ファ」  
 $T(n) \bmod 7 = 4$  なら n番目の音は「ソ」  
 $T(n) \bmod 7 = 5$  なら n番目の音は「ラ」  
 $T(n) \bmod 7 = 6$  なら n番目の音は「シ」

よって

$T(0)=70$  なら 1番目の音は「ド」  
 $T(1)=71$  なら 2番目の音は「レ」  
 $T(2)=72$  なら 3番目の音は「ミ」

...

$T(99)=350$  なら 100番目の音は「ド」

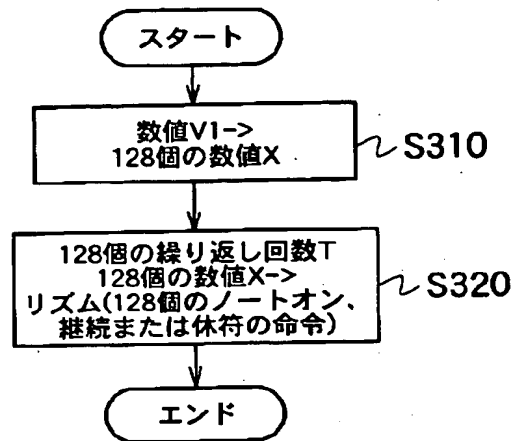
...

$T(125)=144$  なら 126番目の音は「ソ」  
 $T(126)=215$  なら 127番目の音は「ラ」  
 $T(127)=288$  なら 128番目の音は「シ」

【図8】

- (a) 座標点(SX+DX, SY+DY)について繰り返し回数Tを求める  
 $Z(0) = (SX+DX) + (SY+DY)$   
 $Z(1) = Z(0)^2 + Z(0)$   
 $Z(2) = Z(1)^2 + Z(0)$   
 :
- (b) 座標点(SX+2\*DX, SY+2\*DY)について繰り返し回数Tを求める  
 $Z(0) = (SX+2*DX) + (SY+2*DY)$   
 $Z(1) = Z(0)^2 + Z(0)$   
 $Z(2) = Z(1)^2 + Z(0)$   
 :  
 座標点(SX+127\*DX, SY+127\*DY)について繰り返し回数Tを求める  
 $Z(0) = (SX+127*DX) + (SY+127*DY)$   
 $Z(1) = Z(0)^2 + Z(0)$   
 $Z(2) = Z(1)^2 + Z(0)$   
 :  
 :

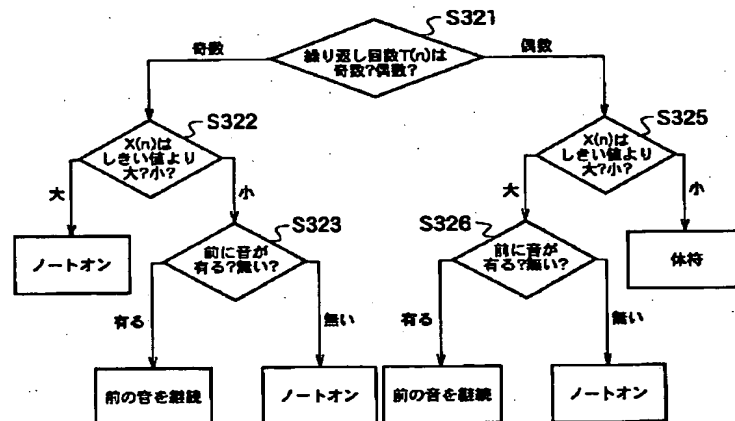
【図11】



【図12】

- (a)  $X(0) = 0.01 + 0.98 * V1 / 65535$   
 $X(1) = \lambda * X(0) * (1 - X(0))$   
 $X(2) = \lambda * X(1) * (1 - X(1))$   
 :  
 $X(126) = \lambda * X(125) * (1 - X(125))$   
 $X(127) = \lambda * X(126) * (1 - X(126))$   
 :
- (b)  $X(0) = 0.01 + 0.98 * V1 / 65535$   
 $X(1) = \lambda * X(0) * (1 - X(0))$   
 $X(2) = \lambda * X(1) * (1 - X(1))$   
 :  
 $X(N) = \lambda * X(N-1) * (1 - X(N-1))$   
 $X(N+1) = \lambda * X(N) * (1 - X(N))$   
 $X(N+2) = \lambda * X(N+1) * (1 - X(N+1))$   
 :  
 $X(N+126) = \lambda * X(N+125) * (1 - X(N+125))$   
 $X(N+127) = \lambda * X(N+126) * (1 - X(N+126))$

【図13】



## 【図14】

CBFGECDBFCFDGCOFAADC8DEFG

(C=ド、D=レ、E=ミ、F=ファ、G=ソ、A=ラ、B=シ)

0010200201022101121012010

(0=ノートオン、1=離脱、2=体符)

CB-G\_CD\_F-F\_\_CD--\_DC-\_E-G

つまり、

C 16分音符

B 8分音符

G 16分音符

16分体符

C 16分音符

D 16分音符

16分体符

F 8分音符

F 16分音符

8分体符

C 16分音符

D付点8分音符

16分体符

D 16分音符

C 8分音符

16分体符

E 8分音符

G 16分音符

フロントページの続き

(72)発明者 田中 亀義

東京都新宿区市谷薬王寺町58春日マンション4F 株式会社エム・ゾーン内

Fターム(参考) 5D378 MM67 PP01 PP03